

## What is React?

- i) React is a JavaScript library for building user interfaces (Not Framework).
- ii) It is used to build single-page applications.
- iii) React allows us to create reusable UI components.

Frame Works  
(House bought)

UI Library  
(House built by yourself)

### Similarity

Both the pieces of code written to solve complicated problems.

Parameters	Library	Framework
Definition	Libraries provide developers with predefined classes and functions to make their work easier and boost the development process.	Framework is like the foundation upon which developers build applications for specific platforms.
Inversion of Control	By using a library, you can control the flow of the application and call the library.	The framework controls the flow and calls your code for any custom behaviour.

<u>Collections</u>	Generally, libraries are a collection of helper modules, objects, classes, functions, message templates, pre-written code, etc.	Frameworks consist of a lot of APIs, compilers, toolsets, support programs, libraries, etc.
Code Modification	Code in libraries are geared towards a particular program or to solve a specific development problem. Therefore, developer must modify library code to meet their needs.	Despite the fact that frameworks generates new codes for developers. These codes can't be altered or modified later. Unlike libraries, frameworks don't allow users to modify their pre-written codes, so you don't have to worry about deleting or changing them.
Scope	It is possible to call a library out of context. You may use the library wherever you see fit in your code.	You can only call and use what belongs to a framework within the same framework.
Function	In the program linking and binding process, they play an important role.	Using them, you can build and deploy applications in a standard way as the framework already provides code to perform common tasks and uses code provided by a developer for custom functionality.

## Compared to

Having a library means understanding the functionality of each method, and it is not easy to create complex interactions since you need to call many methods to get the desired results.

## Extensibility

Generally, libraries aren't designed for extensibility. They are designed to accomplish a specific purpose.

## Replaceable

It is easy to replace library with another library.

## Performance

Less code is required to build libraries, which leads to faster loading times and better performance.

## Usage

The purpose of libraries is to perform a defined and specific task e.g.) Image manipulation, Network protocols, math operations, etc.

## Existing Project

Libraries can be integrated seamlessly into existing projects to add functionality.

## Example

JQuery, React.js, etc.

Frameworks provide a basic plan, but some plugins need to be added to code, it is easier to do the right modifications.

Frameworks provide general functionality. Because of this, they are built to be extensible, which allows developers to incorporate app-specific features without modifying the framework's source code.

Frameworks are difficult to replace.

Developing a framework requires a lot of coding, which increases loading time and decreases performance.

Performs wide range of tasks. e.g.) Web - Application system, plug-in managers, GUI systems, and so on.

Incorporating frameworks seamlessly into an existing project is impossible.

Spring, Node.js, Angular.js, Vue.js, etc.

## Content Delivery Network (CDN)

- CDN is a network of interconnected servers that speeds up webpage applications.
- When a user visits a website, data from the website's server has to travel across the internet to reach the user's computer. If a user is located far from that server, it will take a long time to load a large file such as video or website image. Instead, if website content is stored on CDN servers geographically closer to the user and reaches their computer much faster.

## Importance of CDN

- A CDN improves efficiency by introducing intermediate servers between client and website server.
- These CDN servers manage some of the traffic to the web server, decrease bandwidth consumption, and expand communication. They reduce bandwidth usage of web server, reduce user experience of year applications.

## Benefits of CDNs

- i) Reduce page load time.
- ii) Reduce bandwidth costs.
- iii) Increases content availability.
- iv) Improves website security.

What internet content can a CDN deliver?

Two types of content:-

- i) static content
- ii) dynamic content.

## Working of CDN

- CDN work by establishing a point of presence (POP) or a group of CDN edge servers at multiple geographical locations.
- These geographically distributed network nodes work on the principles of caching, dynamic acceleration, and edge logic computations.

## Caching

- Caching is the process of storing ~~entering~~ <sup>multiple copies</sup> fewer data of the same data for access.
- In CDN technology, the term refers to the process of storing static website content on multiple servers in the network.

## Caching in CDN usage as follows

- 1) A geographically remote website visitor makes the first request for static web content from your site.
- 2) The request reaches your web application server or origin server. The origin server sends the response to the remote visitor. At the same time, it also sends a copy of the response to the CDN POP geographically close to that visitor. CDN POP stores the copy as a cached file.
- 3) The next time the visitor, or any other visitor in the same location, makes the same request, the caching server, not the origin server, sends the response.

## Dynamic Acceleration

- Dynamic acceleration is the reduction in response time for dynamic web content requests because of an intermediary CDN server b/w the web application and the client.

## Edge Logic Computation

- One can perform the CDN edge server to perform logical computations that simplify communication b/w the client and the server.
- b/w the client and the server can do the following:-
- For eg, this server can modify caching behaviors.
  - i) Inspect user requests and handle incorrect user requests.
  - ii) Validate and handle content before responding.
  - iii) Modify or optimize content before responding.

## Uses of CDN

- High speed content delivery.
- ii) Real-time streaming.
- iii) Multi-user scaling.

- Q Why is React known as "React" because of its feature, which is ability to "react" or respond dynamically to changes in data.

- The name "React" also reflects the fact that it builds around the concept of data flow, where changes in data triggers updates through component hierarchy, helping to avoid re-renders as necessary. This approach helps to avoid common issues with two-way data binding, such as performance problems and hard-to-debug code.

- Q1 What is cross origin in script tag? The cross origin attribute sets the mode of execution. The cross origin attribute sets the mode of execution of request to an HTTP CORS Request.
- A cross-origin request is a request for a resources (stylesheets, scripts, fonts, images) from another domain.
  - CORS is used to manage cross-origin requests.
  - CORS stands for cross-origin Resource sharing, and is a mechanism that allows resources on a webpage to be requested from another domain to be requested from another domain.
  - It defines a way of how a browser and server can interact to determine whether it is safe to allow the cross-origin request.
  - CORS allows servers to specify who can access the things.

## React V15 React DOM

- React is used for building reusable UI components and managing their state, while React is closer to render those components to DOM.
- `react.development.js`
- |  |  |
|--|--|
| → intended to be used during development phase of web application  | → used in production ready web applications.   |
| → it is not optimized for performance and includes additional warnings and error messages in the console that helps developers identify potential issues with their application. | → it is optimized for performance and doesn't include the additional warnings and error messages found in the development version. |

Asynchronous and defer attributes in JS

Different ways to load JS:

1) Inline script

```
<script>  
    alert("Hello, world!");  
</script>
```

2) External script

```
<script src="script.js"></script>
```

3) Aynchronous script

```
<script src="script.js" async></script>
```

• allows HTML page to continue rendering while the script loads in the background, e.g.,

```
<script src="script.js" asyn></script>
```

4) Defer script

→ defers loading of a script until after the HTML document has finished parsing by setting the defer attribute to true.

```
<script src="script.js" defer></script>
```

5) Dynamic script loading

```
const script = document.createElement('script');  
script.src = 'script.js';
```

document.head.appendChild(script);

→ allows you to load scripts based on user interactions or other events.

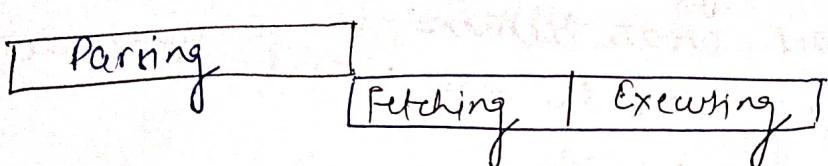
Best case case of these different ways!

### 1) Inline script

→ best for small scripts or scripts that are specific to a single page.

### 2) External script

→ best for larger scripts or scripts that are shared across multiple pages.



[ Parsing Again ]

### 3) Asynchronous script

→ best for scripts that don't need to be executed immediately.

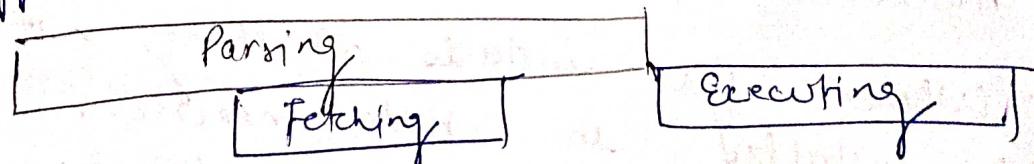
→ allows the page to continue rendering while the scripts load in the background, which can improve the initial page load time.

→ the order of execution of async scripts is not guaranteed, which can lead to unexpected behaviour (Refrain from using in dependent scripts).

### 4) Script

→ best among all

→ allows the page to continue rendering while the scripts load in the background, but ensures that the scripts are executed in the order they appear in the document.



### 5) Dynamic script loading

→ allows you to load scripts on-demand which can reduce the initial page load time.